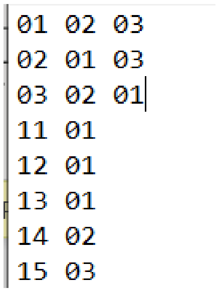
Network setup

Use a central control to setup the network topology. The control reads the config file to find the topology. When initialing peers, supper peer knows the neighbor supper peers. Leaf node knows the super peer.



```
01 02 03
02 01 03
03 02 01
11 01
12 01
13 01
14 02
15 03
```

For all-to-all topology, peer 01 is a super peer. It has neighbor supper peers "02 03".

Peer 11 is a leaf node. Its super peer is "peer 01".

Peers

Data Structures

`HashMap <String, Integer> masterFile` Monitoring the version number of files in the master folder

`HashMap <String, ArrayList<String>> downloadFile;` Monitoring the version number of files in the download folder

`HashMap <String, ArrayList<String>> fileStat;` Monitoring the file status on the super peer.

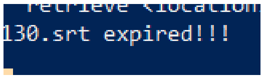
The array list stores the following information `[filename, versionNum, consistant, originID, download, TTR]`

Push

A peer can only modify its files in the "master" file fold. When a file is modified using the function "modify(filename)". A invalidate message will be generated and send to its super peer. The super peer will remove the cached file name from the file index and broadcast to the neighbors. All the super peer having the file would remove it from its file index.

Poll

We implemented a thread poll function at super peer. The thread poll function will periodically check its fileStat. Each time, it will decrease the TTR of each downloaded file. If the TTR is below zero, it notifies the use that the file is expired.



```
retrieve <location
130.srt expired!!!
```

On each leaf node, we can use the function "refresh" to obtain new files of those expired files.

Testing

A function “ranSearch” was implemented to randomly send a query searching for a file.

“ranM” will randomly modify a file.